**使用MapReduce求每年最低温度**

姓名：刘健辉

学号：202083013

（注意：在下面文档标黄处加入要求的代码或结果截图，不要修改原文档。）

**实验原理**

实验数据源内容如下所示：

0029028060999991908010106004+69100+027217FM-12+014999999V0209991C000019999999N0000001N9-02441+99999100261ADDGF104991999999999999999999

其中第16-19个字符代表对应的年份；第88个字符代表当次采集到的温度的符号位，第89-92个字符代表当次采集到的温度绝对值（如果为9999则代表无效值），例如-0244则代表零下24.4℃；第93个字符代表当次采集值是否有效（如果为01459中的值，则代表是有效值）。

根据上述分析，可以清晰的制定MapReduce程序的流程。在Map阶段完成年份以及有效温度的筛选工作，Shuffle阶段完成相同年份数据的聚集和分发工作（这个过程是MapReduce的默认过程，不用具体配置），Reduce阶段负责接收所有年份数据并计算最低温度值。

**实验步骤**

1  **启动Hadoop集群**

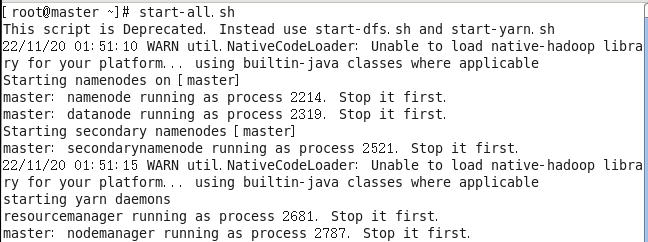
在进行实验之前，首先需要确保Hadoop集群已经正常启动。如果Hadoop没有启动，需要通过下列步骤启动Hadoop集群。

**在模板中，我们已经配置好了Hadoop伪分布式环境，同学们不需要再次配置，可以直接启动使用。**

**步骤1.**  **启动Hadoop**

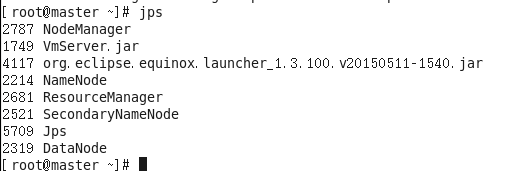
打开一个终端模拟器，启动Hadoop。

代码及结果请截图放在此处：



**步骤2.**  **验证Hadoop是否启动成功**

代码及结果请截图放在此处：

****

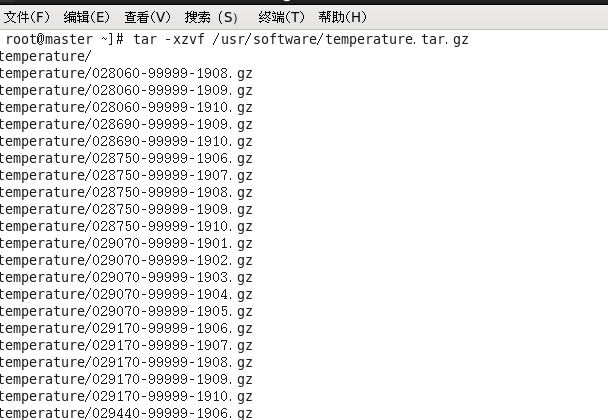
2  **数据准备**

在模板中已经预制好相关数据源temperature.tar.gz，存放于/usr/software/目录下，我们可以直接解压使用。

**步骤1.**  **数据提取**

1. 通过命令解压气象数据集。

代码及结果请截图放在此处：



2.   解压完成后会生成temperature文件夹，进入到temperature文件夹并使用zcat命令把这些数据文件解压合并到sample.txt文件中。

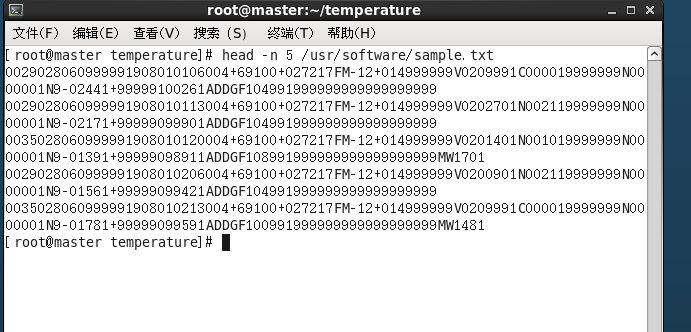
cd temperature

zcat \*.gz > /usr/software/sample.txt

3.   通过下面的命令查看数据是否提取成功。

head -n 5 /usr/software/sample.txt

结果请截图放在此处：

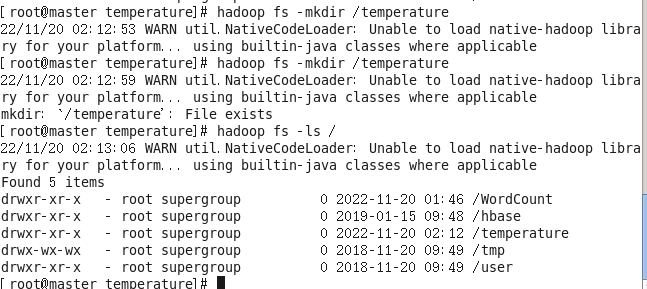


**步骤2.**  **数据源上传**

数据提取成功后，我们需要进一步将数据上传到HDFS中，以作为MapReduce程序的输入文件。

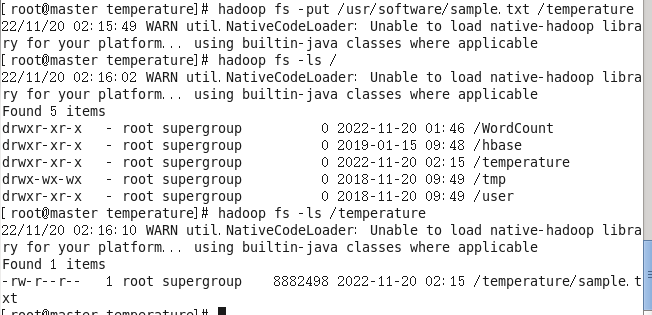
1. 在HDFS中创建工作目录/temperature。

代码及结果请截图放在此处：



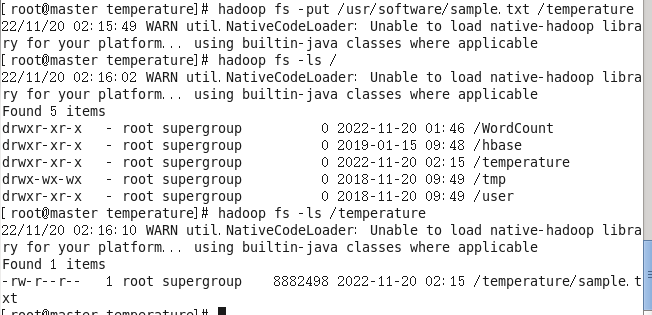
1. 将数据文件sample.txt上传到工作目录。

代码及结果请截图放在此处：



1. 查看文件是否上传成功。

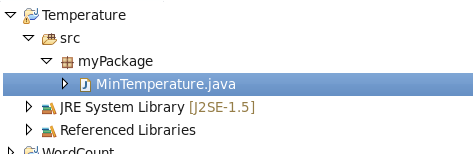
代码及结果请截图放在此处：



3  **项目工程创建**

打开Eclipse，创建工程Temperature并导入相关的Hadoop依赖包，在Temperature工程中创建一个package命名为myPackage，同时在myPackage中创建类文件MinTemperature。

将创建的工程结构截图放在此处：（注意：与Hadoop相关的所有依赖包一定要正确导入，否则后文的MapReduce程序将无法正确运行。可参照WordCount实验中的包导入说明。）



4  **类文件编写**

在创建的MinTemperature类文件中写入代码。

程序在Map阶段接收输入的<key, value>(key是当前输入的行号，value是对应行的内容)，然后对此行内容进行筛选，筛选出年份以及对应的采集温度值。

在Reduce阶段，会接收到<year,{value1,value2,value3…}>形式的数据，也就是特定年份及其采集到的温度值，最后组织成<year,min(value)>的形式直接输出。

以下代码中，请编写完成map函数与reduce函数部分。

可能用到的JAVA函数：

（1）substring(int beginIndex, int endIndex)：返回字符串的子字符串。

（2）charAt(int index) ：返回指定索引处的字符

（3）parseInt(String s): 返回字符串s用十进制参数表示的整数值。

（4）int min(int arg1, int arg2)：返回两个参数中的最小值

package myPackage;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.FloatWritable;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

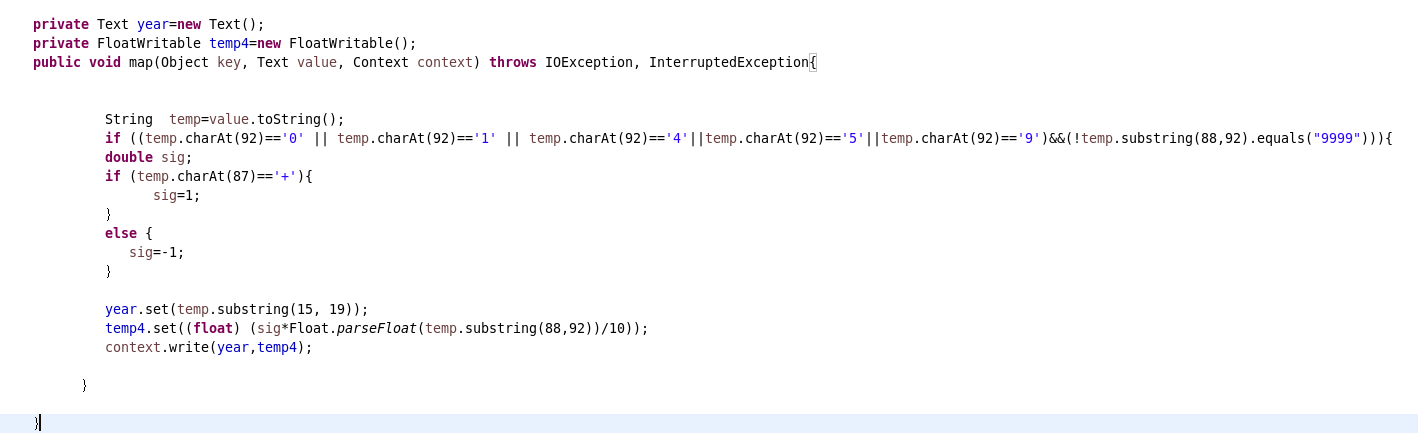
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MinTemperature {

    public static class MinTemperatureMapper extends Mapper<LongWritable, Text, Text, FloatWritable>{

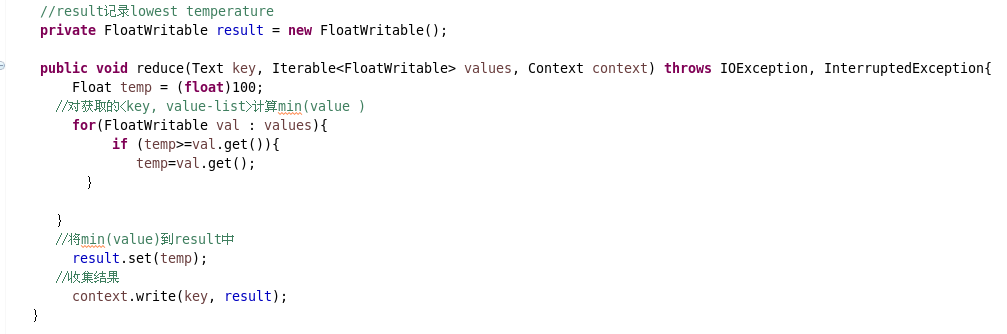
   //请往下编写map函数



}

public static class MinTemperatureReducer extends Reducer<Text, FloatWritable, Text, FloatWritable>{

//请往下编写reduce函数



}

    public static void main(String[] args) throws Exception{

         if(args.length != 2){

             System.err.println("USage: MinTemperature<input path><output path>");

             System.exit(-1);

         }

         Configuration configuration=new Configuration();

         configuration.set("fs.defaultFS", "hdfs://master:9000");

         Job job = new Job(configuration);

         job.setJarByClass(MinTemperature.class);

         job.setMapperClass(MinTemperatureMapper.class);

         job.setReducerClass(MinTemperatureReducer.class);

         job.setJobName("Min temperature");

         FileInputFormat.addInputPath(job, new Path(args[0]));

         FileOutputFormat.setOutputPath(job, new Path(args[1]));

         job.setOutputKeyClass(Text.class);

         job.setOutputValueClass(FloatWritable.class);

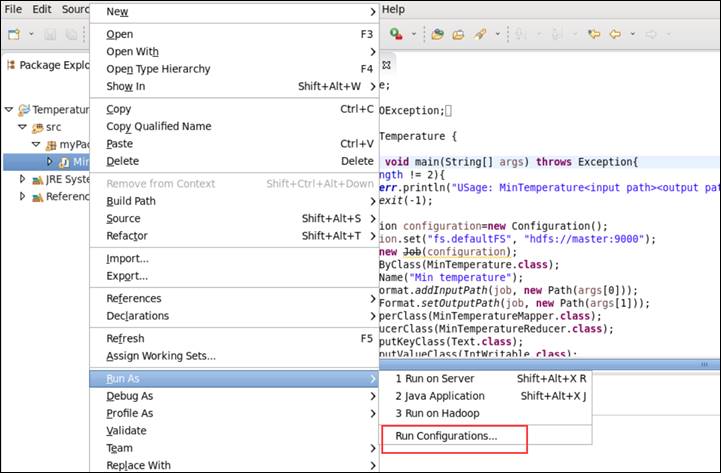
         System.exit(job.waitForCompletion(true)?0:1);

    }

}

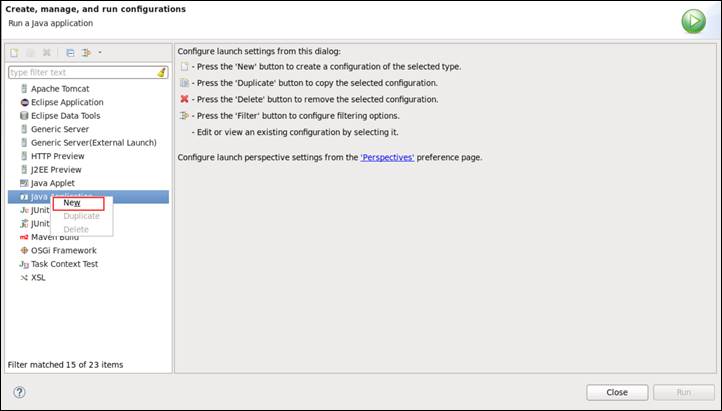
5  **运行程序**

1.   将鼠标移动到类文件MinTemperature上，点击鼠标右键，进行运行环境配置：



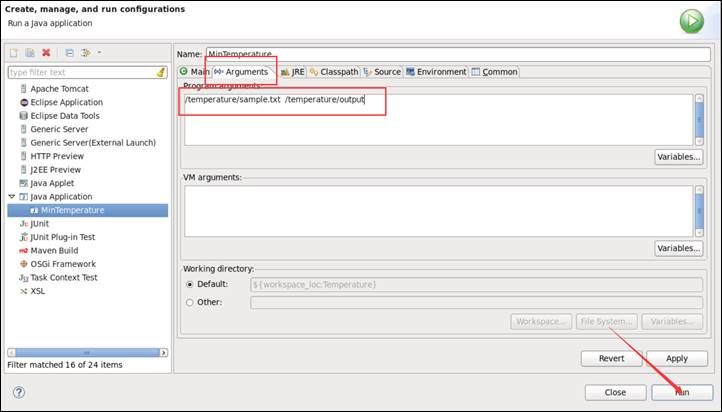
**环境配置**

2.   在“Java Application”选项上点击鼠标右键，选择“New”，此时会自动显示出我们所创建的MinTemperature，如下图所示：



**程序识别**

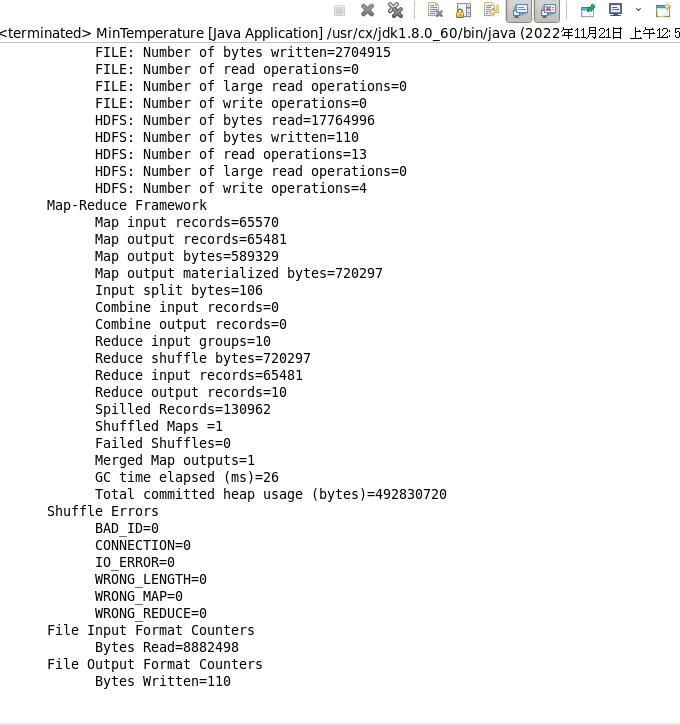
3.   在“Arguments”中添加输入文件的绝对路径以及结果输出目录（中间用空格分隔），其中输入文件的绝对路径为/temperature/sample.txt，结果输出目录为/temperature/output，然后点击“Run”按钮：



**参数配置**

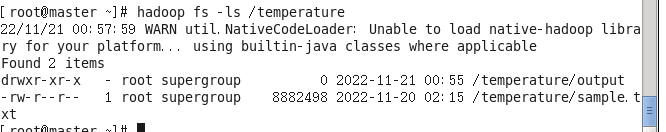
4.   在主界面的“Console”子菜单中，可以实时查看到程序的运行结果，

结果请截图放在此处：



5.   打开终端模拟器，通过下面的命令在HDFS中查看结果输出目录是否生成：

代码及结果请截图放在此处：



6.   MapReduce默认会将结果存储到生成目录下的part-r-00000文件中，可以通过下面的命令进一步查看所生成的文件的内容：

代码及结果请截图放在此处：

